ELSEVIER

Journal of Computational and Applied Mathematics 104 (1999) 193–212

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Comparison of parallel implementation of some multi-level Schwarz methods for singularly perturbed parabolic problems

V. Sirotkin[a], P. Tarvainen[b,*]

[a]*Institute of Microelectronics Technology, Russian Academy of Sciences, Moscow District, 142432, Chernogolovka, Russia*

[b]*Laboratory of Scientific Computing, Department of Mathematics, University of Jyväskylä, P.O. Box 35, FIN-40351 Jyväskylä, Finland*

Received 3 February 1998; accepted 3 December 1998

Abstract

Parallel multi-level algorithms combining a time discretization and an overlapping domain decomposition technique are applied to the numerical solution of singularly perturbed parabolic problems. Two methods based on the Schwarz alternating procedure are considered: a two-level method with auxiliary "correcting" subproblems as well as a three-level method with auxiliary "predicting" and "correcting" subproblems. Moreover, modifications of the methods using time extrapolation on subdomain interfaces are investigated. The emphasis is given to the description of the algorithms as well as their computer realization on a distributed memory multiprocessor computer. Numerical experiments illustrate the performance of the algorithms on parallel environment and their behaviour with respect to the critical parameters, such as the perturbation parameter and the size of the auxiliary subdomains.

Keywords: Singularly perturbed parabolic problem; Multi-level Schwarz method; Parallel computing

1. Introduction

We are interested in parallel algorithms based on overlapping domain decomposition for the numerical solution of the following parabolic problem:

$$\begin{aligned} &\mathcal{L}u(P,t) - \frac{\partial u(P,t)}{\partial t} = f(P,t),\ P=(x,y), \quad (P,t) \in \Omega \times (0,T], \\ &u(P,t) = \bar{u}(P,t),\ (P,t) \in \partial\Omega \times (0,T], \\ &u(P,0) = u^0(P),\ P \in \bar{\Omega}, \end{aligned} \tag{1}$$

* Corresponding author. Present address: Numerola Ltd., Väinönkatu 6, FIN-40100 Jyväskylä, Finland.
E-mail addresses: sirotkin@ipmt-hpm.ac.ru (V. Sirotkin), pasi.tarvainen@numerola.fi (P. Tarvainen).

0377-0427/99/$ - see front matter
PII: S0377-0427(99)00057-6

where $\mathscr{L} \equiv \mu^2(\partial^2/\partial x^2 + \partial^2/\partial y^2)$ is a second-order differential operator with μ being a positive parameter, and $\partial\Omega$ is the boundary of Ω. The functions $f(P,t), \bar{u}(P,t)$ and $u^0(P)$ are assumed to be sufficiently smooth [4,21]. Under suitable continuity and compatibility conditions on the data, a unique solution $u(P,t)$ of problem (1) exists (see [10] for details). For $\mu \ll 1$, problem (1) is singularly perturbed and has parabolic layers near $\partial\Omega \times (0,T]$ (see, e.g., [18]).

Recent evolution of multiprocessor computers has rekindled interest in the development of domain decomposition methods for the numerical solution of partial differential equations (PDEs). In these methods, the original problem is reduced to a family of smaller problems on subdomains which can be solved concurrently. Many of the domain decomposition methods for elliptic PDEs are based on the classical Schwarz alternating procedure [16]. In [11], this algorithm was first extended successfully to a class of parabolic PDEs. Nowadays, a number of methods using the Schwarz procedure has been proposed for the numerical solution of parabolic problems (including singularly perturbed problems) as well as algebraic systems arising from mesh discretization of these problems, see, for example, [6–9,14,17], and references therein. Note that most of these methods are designed for solving elliptic problems resulting from a time discretization of original parabolic problems.

In this paper, we evaluate computational effectiveness of parallel multi-level Schwarz methods described originally in [3,20,21]: a two-level method with auxiliary "correcting" subproblems and a the three-level method with auxiliary "predicting" and "correcting" subproblems. These methods are specifically constructed for parabolic problems. They are based on coupling a time discretization (here, we shall use the backward Euler scheme) with an overlapping domain decomposition technique, so that on each time step only one iteration of the methods is performed. These methods originate from the two-level Schwarz method which was successfully applied to singularly perturbed elliptic problems in [2,19,22,24]. Furthermore, as in [4,20,21], we consider a modification of the multi-level Schwarz methods, where the extrapolation in the time variable leant on the solutions from some previous time steps is applied to the determination of the boundary conditions on a part of subdomain interfaces. Here, we shall examine the modified methods with "two-step" extrapolation (see also [20,21]).

The plan for the rest of this paper is as follows: In Section 2, we describe the multi-level Schwarz methods: the two-level method in Section 2.2, the three-level method in Section 2.3, and the multi-level methods with time extrapolation in Section 2.4. In Section 3, parallel implementation of the methods on a distributed memory computer is proposed. The main emphasis is given to the numerical results which are reported in Section 4. The paper is concluded with some remarks in Section 5.

2. Multi-level Schwarz methods

In this section, we describe the domain decomposition methods for problem (1) and survey with references their theoretical properties.

2.1. Semi-discrete version of the original problem

To construct parallel methods for the iterative solution of problem (1), we shall combine the overlapping Schwarz technique and a time discretization scheme. Approximating the partial derivative

$\partial u/\partial t$ by a backward difference formula, we introduce the following well-known semi-discrete version of problem (1) (the backward Euler scheme):

$$\begin{aligned} &\mathscr{L}U^n(P) - \frac{U^n(P) - U^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \Omega, \\ &U^n(P) = \bar{u}(P,t^n), \quad P \in \partial\Omega, \\ &U^0(P) = u^0(P), \quad P \in \Omega, \\ &t^n = n\tau, \quad 1 \leqslant n \leqslant N, \quad \tau = \frac{T}{N}, \end{aligned} \tag{2}$$

where $\tau > 0$ is the step of time discretization (time step). The following result is known (see, e.g., [15]):

Lemma 1. *If the solution of problem* (1) *is smooth enough, then the following estimate is valid:*

$$\max_{1 \leqslant n \leqslant N} \left[\max_{P \in \Omega} |u(P,t^n) - U^n(P)| \right] \leqslant c\tau,$$

where $u(P,t)$ *and* $U^n(P)$ *are the solutions of problems* (1) *and* (2), *respectively, and the constant* c *is independent of the time step* τ.

2.2. Two-level Schwarz method with correcting subproblems

In order to give an exact description of the method, we first define the necessary notations and assumptions. For simplicity, we here assume that the domain Ω as well as its subdomains are rectangles.

Let Ω be a rectangle $(0,X) \times (0,Y)$. We introduce a *nonoverlapping* multi-domain decomposition of the domain Ω into $K \times L$ subdomains $\Omega_{k,l}, 1 \leqslant k \leqslant L, 1 \leqslant l \leqslant L$, the *first-level subdomains*, as follows:

$$\Omega_{k,l} = (x_{k-1}^C, x_k^C) \times (y_{l-1}^C, y_l^C), \quad 1 \leqslant k \leqslant K, \ 1 \leqslant l \leqslant L,$$

where (see Fig. 1)

$$0 < x_{k-1}^C < x_k^C < X, \quad 2 \leqslant k \leqslant K-1, \quad x_0^C = 0, \quad x_K^C = X,$$

$$0 < y_{l-1}^C < y_l^C < Y, \quad 2 \leqslant l \leqslant L-1, \quad Y_1^C = 0, \quad y_L^C = Y.$$

Furthermore, we define sets of interface subdomains $\{\xi_{k,l}^x\}, \{\xi_{k,l}^y\}$ and $\{\xi_{k,l}^{xy}\}$, the *second-level subdomains*, by (see Fig. 1):

$$\xi_{k,l}^x = (x_k^b, x_k^e) \times (y_{l-1}^C, y_l^C), \quad 1 \leqslant k \leqslant K-1, \ 1 \leqslant l \leqslant L,$$

$$\xi_{k,l}^y = (x_{k-1}^C, x_k^C) \times (y_l^b, y_l^e), \quad 1 \leqslant k \leqslant K, \ 1 \leqslant l \leqslant L-1,$$

$$\xi_{k,l}^{xy} = (x_k^b, x_k^e) \times (y_l^b, y_l^e), \quad 1 \leqslant k \leqslant K-1, \ 1 \leqslant l \leqslant L-1,$$

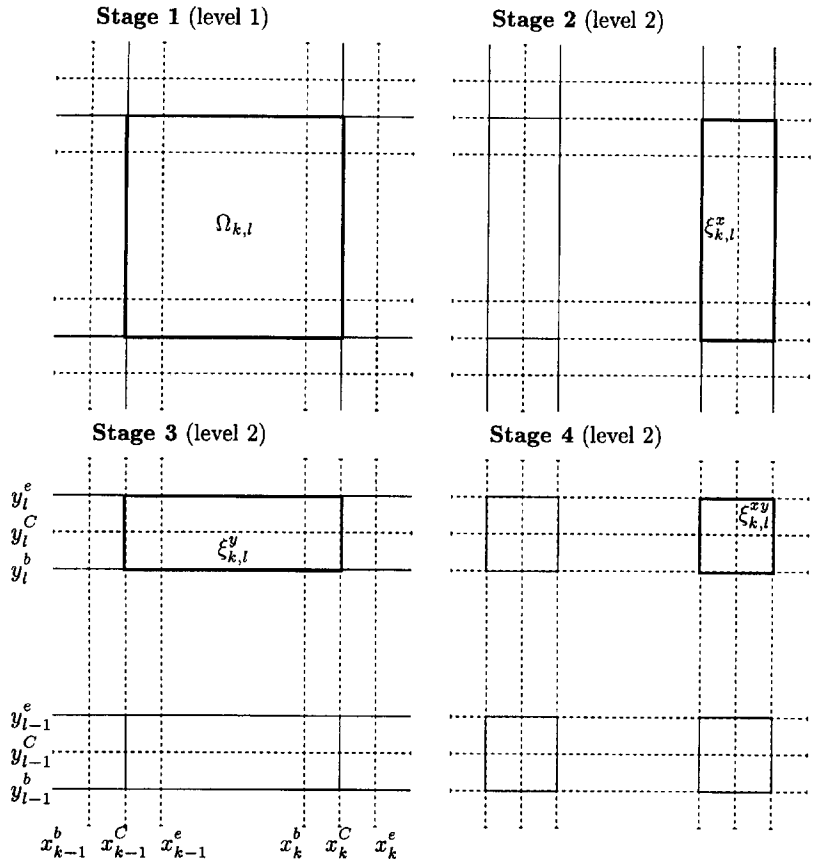

Fig. 1. Schematic representation of the two-level Schwarz method (Algorithm A1) realized on $K \times L$ processors: One processor (numbered, e.g., $(l-1)\times K + k$) solves sequentially the subproblem associated with the first-level subdomain $\Omega_{k,l}$ and three correcting subproblems related to the second-level subdomains $\xi^x_{k,l}, \xi^y_{k,l}$ and $\xi^{xy}_{k,l}$.

$$0 < x^b_k < x^C_k < x^e_k < X, \quad 1 \leqslant k \leqslant K-1, \quad 0 < y^b_l < y^C_l < y^e_l < Y, \quad 1 \leqslant l \leqslant L-1,$$

$$\Xi^x = \bigcup_{\substack{1\leqslant k\leqslant K-1\\ 1\leqslant l\leqslant L}} \xi^x_{k,l}, \qquad \Xi^y = \bigcup_{\substack{1\leqslant k\leqslant K\\ 1\leqslant l\leqslant L-1}} \xi^y_{k,l}, \qquad \Xi^{xy} = \bigcup_{\substack{1\leqslant k\leqslant K-1\\ 1\leqslant l\leqslant L-1}} \xi^{xy}_{k,l}.$$

The overlapping interval sizes of the first-level and second-level subdomains are determined via the following parameters (see Fig. 1):

$$\begin{aligned} d^x_{1,k} &= x^C_k - x^b_k, \qquad d^x_{2,k} = x^e_k - x^C_k, \qquad 1 \leqslant k \leqslant K-1; \\ d^y_{1,l} &= y^C_l - y^b_l, \qquad d^y_{2,l} = y^e_l - y^C_l, \qquad 1 \leqslant l \leqslant L-1. \end{aligned} \tag{3}$$

Construction of the proposed domain decomposition method is based on the semi-discrete version (2) of problem (1) presented in Section 2.1. In the method, all the subproblems on the first-level subdomains are solved concurrently. These solutions are then corrected using auxiliary interface

problems which are defined on the second-level subdomains. Thus, the two-level Schwarz method with correcting subproblems consists of the following stages:

Algorithm A1: Two-level Schwarz method

Stage 0. Initialization: $W^0(P) = u^0(P), P \in \overline{\Omega}$.

Stage 1. Solve *simultaneously* the subproblems on the *first-level* subdomains $\Omega_{k,l}, 1 \leqslant k \leqslant K$, $1 \leqslant l \leqslant L$:

$$\begin{aligned} &\mathscr{L} v^n_{k,l}(P) - \frac{v^n_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P, t^n), \quad P \in \Omega_{k,l}, \\ &v^n_{k,l}(P) = \tilde{u}(P, t^n), \quad P \in \partial\Omega_{k,l} \cap \partial\Omega, \\ &v^n_{k,l}(P) = W^{n-1}(P), \quad P \in \partial\Omega_{k,l} \setminus \partial\Omega, \end{aligned} \tag{4}$$

where

$$W^n(P) = \begin{cases} V^n(P), & P \in \Omega_{k,l} \setminus (\overline{\Xi}^x \cup \overline{\Xi}^y), \\ \overset{xy}{\Psi}{}^n(P) & P \in \overline{\Xi}^{xy}, \\ \overset{x}{\Psi}{}^n(P), & P \in \overline{\Xi}^x \setminus \overline{\Xi}^{xy}, \\ \overset{y}{\Psi}{}^n(P) & P \in \overline{\Xi}^y \setminus \overline{\Xi}^{xy}, \end{cases} \qquad n \geqslant 1; \tag{5}$$

$$\begin{aligned} &V^n(P) = v^n_{k,l}(P), \quad P \in \overline{\Omega}_{k,l}, \quad 1 \leqslant k \leqslant K, \ 1 \leqslant l \leqslant L, \\ &\overset{x}{\Psi}{}^n(P) = \overset{x}{\psi}{}^n_{k,l}(P), \quad P \in \overline{\xi}^x_{k,l}, \quad 1 \leqslant k \leqslant K-1, \ 1 \leqslant l \leqslant L, \\ &\overset{y}{\Psi}{}^n(P) = \overset{y}{\psi}{}^n_{k,l}(P), \quad P \in \overline{\xi}^y_{k,l}, \quad 1 \leqslant k \leqslant K, \ 1 \leqslant l \leqslant L-1, \\ &\overset{xy}{\Psi}{}^n(P) = \overset{xy}{\psi}{}^n_{k,l}(P), \quad P \in \overline{\xi}^{xy}_{k,l}, \quad 1 \leqslant k \leqslant K-1, \ 1 \leqslant l \leqslant L-1. \end{aligned}$$

Stage 2. Solve *simultaneously* the subproblems on the *second-level* subdomains $\xi^x_{k,l}, 1 \leqslant k \leqslant K-1$, $1 \leqslant l \leqslant L$:

$$\begin{aligned} &\mathscr{L} \overset{x}{\psi}{}^n_{k,l}(P) - \frac{\overset{x}{\psi}{}^n_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P, t^n), \quad P \in \xi^x_{k,l}, \\ &\overset{x}{\psi}{}^n_{k,l}(P) = V^n(P), \quad P \in \partial\xi^x_{k,l}. \end{aligned} \tag{6}$$

Stage 3. Solve *simultaneously* the subproblems on the *second-level* subdomains $\xi^y_{k,l}, 1 \leqslant k \leqslant K$, $1 \leqslant l \leqslant L-1$:

$$\begin{aligned} &\mathscr{L} \overset{y}{\psi}{}^n_{k,l}(P) - \frac{\overset{y}{\psi}{}^n_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P, t^n), \quad P \in \xi^y_{k,l}, \\ &\overset{xy}{\psi}{}^n_{k,l}(P) = V^n(P), \quad P \in \partial\xi^y_{k,l}. \end{aligned} \tag{7}$$

Stage 4. Solve *simultaneously* the subproblems on the *second-level* subdomains $\xi^{xy}_{k,l}, 1 \leqslant k \leqslant K-1,\ 1 \leqslant l \leqslant L-1$:

$$\begin{aligned} &\mathscr{L}\overset{xy}{\psi}{}^{n}_{k,l}(P) - \frac{\overset{xy}{\psi}{}^{n}_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \xi^{xy}_{k,l}, \\ &\overset{xy}{\psi}{}^{n}_{k,l}(P) = \overset{x}{\Psi}{}^{n}(P), \quad P \in \partial\xi^{xy}_{k,l} \cap \bar{\Xi}^{x}, \\ &\overset{xy}{\psi}{}^{n}_{k,l}(P) = \overset{y}{\Psi}{}^{n}(P), \quad P \in \partial\xi^{xy}_{k,l} \cap \Xi^{y}. \end{aligned} \tag{8}$$

Stage 5. Stopping criterion: If $t^n = T$, then stop; otherwise go to Stage 1.

We emphasize that in (5) the *continuous* function $W^n(P)$, $P \in \Omega$, which is taken as the solution at the nth time step, is obtained in a single pass of Algorithm A1.

Convergence and a convergence rate estimate depending on the perturbation parameter overlapping interval sizes and the time step, for the two-level Schwarz method was proved in [21]:

Theorem 1. *The two-level Schwarz method (Algorithm* A1*) converges to the solution* $U^n(P)$ *of problem* (2) *with linear rate* $q_{A1} \in (0,1)$, *that is*

$$\max_{1 \leqslant n \leqslant N} \left[\max_{P \in \Omega} |U^n(P) - W^n(P)| \right] \leqslant c q_{A1},$$

where the function $W^n(P)$ *was defined by* (5), *and the constant* c *is independent of* q_{A1}. *For* q_{A1}, *the following bound holds:*

$$q_{A1} < 2\left\{ \exp\left[-\frac{D^x}{\mu\tau^{1/2}} \right] + \exp\left[-\frac{D^y}{\mu\tau^{1/2}} \right] \right\},$$

where μ *was given in Section* 1, τ *is the time step introduced in* (2), *and* D^x *and* D^y *are defined by*

$$D^x = \min_{1 \leqslant k \leqslant K-1} \{\min[d^x_{1,k}, d^x_{2,k}]\}, \qquad D^y = \min_{1 \leqslant l \leqslant L-1} \{\min[d^y_{1,l}, d^y_{2,l}]\},$$

with $d^x_{1,k}, d^x_{2,k}$ *and* $d^y_{1,l}, d^y_{2,l}$ *given by* (3).

Remark 1. As is evident from the above description, Algorithm A1 involves four sequential stages. Note that it can be realized on $K \times L$ processors inasmuch all the correcting subproblems can be properly placed into processors containing one of the first-level subproblems from (4). In this situation, each used processor would consist a maximum of four subproblems (for more details, see Fig. 1).

In fact, we could reduce the number of sequential stages of Algorithm A1 to three. Namely, the subproblems from (6) and (7) (in Stages 2 and 3, respectively) can be solved in parallel. In this case, parallel implementation of Algorithm A1 requires at least $(K-1) \times L + K \times (L-1)$ processors (in relation to the total number of the subproblems associated with the subdomains $\xi^x_{k,l}$ and $\xi^y_{k,l}$).

2.3. Three-level Schwarz method with predicting and correcting subproblems

The second variant of the Schwarz method considered in this paper is a three-level iteration process with additional interface subproblems introduced in order to improve the convergence rate

of the resulting algorithm. These (to be called: predicting) subproblems are solved on the zeroth level of the algorithm and from their solutions boundary conditions for the first-level subproblems are determined. The predicting subproblems are defined on the sets of the *zeroth-level subdomains* $\{\theta^x_{k,l}\}$ and $\{\theta^y_{k,l}\}$ which are introduced using the second-level partitioning from Section 2.2 (see Fig. 2):

$$\theta^x_{k,l} = (x^b_k, x^e_k) \times (y^b_{l-1}, y^e_l), \quad 1 \leqslant k \leqslant K-1, \quad 1 \leqslant l \leqslant L,$$

$$\theta^y_{k,l} = (x^b_{k-1}, x^e_k) \times (y^b_l, y^e_l), \quad 1 \leqslant k \leqslant K, \quad 1 \leqslant l \leqslant L-1,$$

$$\Theta^x = \bigcup_{\substack{1 \leqslant k \leqslant K-1 \\ 1 \leqslant l \leqslant L}} \theta^x_{k,l}, \qquad \Theta^y = \bigcup_{\substack{1 \leqslant k \leqslant K \\ 1 \leqslant l \leqslant L-1}} \theta^y_{k,l},$$

with $x^b_0 \equiv 0, x^e_K \equiv X$ and $y^b_0 \equiv 0, y^e_L \equiv Y$.

Based on the given assumptions and notations, the three-level Schwarz method can be presented in the following form:

Algorithm A2: Three-level Schwarz method

Stage 0. Initialization: $W^0(P) = u^0(P)$, $P \in \overline{\Omega}$.

Stage 1. Solve *simultaneously* the subproblems on the *zeroth-level* subdomains $\theta^x_{k,l}$, $1 \leqslant k \leqslant K-1$, $1 \leqslant l \leqslant L$:

$$\begin{aligned} &\mathscr{L}\overset{x}{\phi}{}^n_{k,l}(P) - \frac{\overset{x}{\phi}{}^n_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \theta^x_{k,l}, \\ &\overset{x}{\phi}{}^n_{k,l}(P) = \bar{u}(P,t^n), \quad P \in \partial\theta^x_{k,l} \cap \partial\Omega, \\ &\overset{x}{\phi}{}^n_{k,l}(P) = W^{n-1}(P), \quad P \in \partial\theta^x_{k,l} \setminus \partial\Omega, \end{aligned} \tag{9}$$

where

$$W^n(P) = \begin{cases} V^n(P), & P \in \Omega_{k,l} \setminus (\overline{\Xi}^x \cup \overline{\Xi}^y), \\ \overset{xy}{\Psi}{}^n(P), & P \in \overline{\Xi}^{xy}, \\ \overset{x}{\Psi}{}^n(P), & P \in \overline{\Xi}^x \setminus \overline{\Xi}^{xy}, \\ \overset{y}{\Psi}{}^n(P), & P \in \overline{\Xi}^y \setminus \overline{\Xi}^{xy}, \end{cases} \qquad n \geqslant 1, \tag{10}$$

$$\begin{aligned} &V^n(P) = v^n_{k,l}(P), \quad P \in \overline{\Omega}_{k,l}, \quad 1 \leqslant k \leqslant K, \ 1 \leqslant l \leqslant L, \\ &\overset{x}{\Psi}{}^n(P) = \overset{x}{\psi}{}^n_{k,l}(P), \quad P \in \overline{\xi}^x_{k,l}, \quad 1 \leqslant k \leqslant K-1, \ 1 \leqslant l \leqslant L, \\ &\overset{y}{\Psi}{}^n(P) = \overset{y}{\psi}{}^n_{k,l}(P), \quad P \in \overline{\xi}^y_{k,l}, \quad 1 \leqslant k \leqslant K, \ 1 \leqslant l \leqslant L-1, \\ &\overset{xy}{\Psi}{}^n(P) = \overset{xy}{\Psi}{}^n_{k,l}(P), \quad P \in \overline{\xi}^{xy}_{k,l}, \quad 1 \leqslant k \leqslant K-1, \ 1 \leqslant l \leqslant L-1. \end{aligned}$$

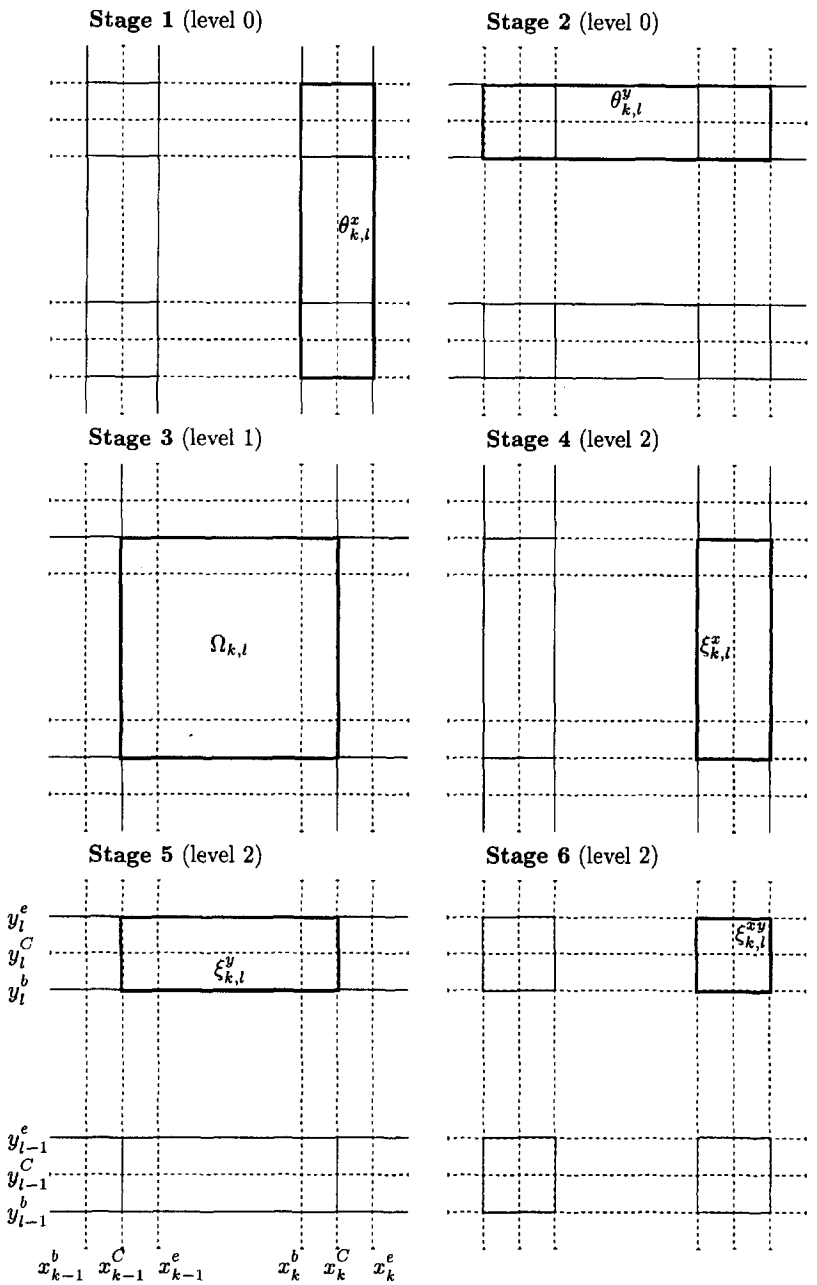

Fig. 2. Schematic representation of the three-level Schwarz method realized on $K \times L$ processors: One processor (numbered, e.g., $(l-1) \times K + k$) solves sequentially two predicting subproblems related to the zeroth-level subdomains $\theta^x_{k,l}$ and $\theta^y_{k,l}$, the subproblem associated with the first-level subdomain $\Omega_{k,l}$, and three correcting subproblems on the second-level subdomains $\xi^x_{k,l}$, $\xi^y_{k,l}$ and $\xi^{xy}_{k,l}$.

Stage 2. Solve *simultaneously* the subproblems on the *zeroth-level* subdomains $\theta^{y}_{k,l}$, $1 \leqslant k \leqslant K$, $1 \leqslant l \leqslant L-1$:

$$\begin{aligned}
&\mathscr{L}\overset{y}{\phi}{}^{n}_{k,l}(P) - \frac{\overset{y}{\phi}{}^{n}_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \theta^{y}_{k,l},\\
&\overset{y}{\phi}{}^{n}_{k,l}(P) = \bar{u}(P,t^n), \quad P \in \partial\theta^{y}_{k,l} \cap \partial\Omega,\\
&\overset{y}{\phi}{}^{n}_{k,l}(P) = W^{n-1}(P), \quad P \in \partial\theta^{y}_{k,l} \setminus \partial\Omega.
\end{aligned} \tag{11}$$

Stage 3. Solve *simultaneously* the subproblems on the *first-level* subdomains $\Omega_{k,l}$, $1 \leqslant k \leqslant K$, $1 \leqslant l \leqslant L$:

$$\begin{aligned}
&\mathscr{L}v^{n}_{k,l}(P) - \frac{v^{n}_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \Omega_{k,l},\\
&v^{n}_{k,l}(P) = \bar{u}(P,t^n), \quad P \in \partial\Omega_{k,l} \cap \partial\Omega,\\
&v^{n}_{k,l}(P) = \overset{x}{\Phi}{}^{n}(P), \quad P \in \gamma^{x}_{k,l},\\
&v^{n}_{k,l}(P) = \overset{y}{\Phi}{}^{n}(P), \quad P \in \gamma^{y}_{k,l},
\end{aligned} \tag{12}$$

where $\gamma^{x}_{k,l} = \partial\Omega_{k,l} \setminus \partial\Omega \cap (\Theta^{x} \setminus \Xi^{y})$, $\gamma^{y}_{k,l} = \partial\Omega_{k,l} \setminus \partial\Omega \cap (\Theta^{y} \setminus \Xi^{x})$, and

$$\begin{aligned}
&\overset{x}{\Phi}{}^{n}(P) = \overset{x}{\phi}{}^{n}_{k,l}(P), \quad P \in \bar{\theta}^{x}_{k,l}, \quad 1 \leqslant k \leqslant K-1, \ 1 \leqslant l \leqslant L,\\
&\overset{y}{\Phi}{}^{n}(P) = \overset{y}{\phi}{}^{n}_{k,l}(P), \quad P \in \bar{\theta}^{y}_{k,l}, \quad 1 \leqslant k \leqslant K, \ 1 \leqslant l \leqslant L-1.
\end{aligned}$$

Stage 4. Solve *simultaneously* the subproblems on the *second-level* subdomains $\xi^{x}_{k,l}$, $1 \leqslant k \leqslant K-1$, $1 \leqslant l \leqslant L$:

$$\begin{aligned}
&\mathscr{L}\overset{x}{\psi}{}^{n}_{k,l}(P) - \frac{\overset{x}{\psi}{}^{n}_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \xi^{x}_{k,l},\\
&\overset{x}{\psi}{}^{n}_{k,l}(P) = V^{n}(P), \quad P \in \partial\xi^{x}_{k,l}.
\end{aligned} \tag{13}$$

Stage 5. Solve *simultaneously* the subproblems on the *second-level* subdomains $\xi^{y}_{k,l}$, $1 \leqslant k \leqslant K$, $1 \leqslant l \leqslant L-1$:

$$\begin{aligned}
&\mathscr{L}\overset{y}{\psi}{}^{n}_{k,l}(P) - \frac{\overset{y}{\psi}{}^{n}_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \xi^{y}_{k,l},\\
&\overset{y}{\psi}{}^{n}_{k,l}(P) = V^{n}(P), \quad P \in \partial\xi^{y}_{k,l}.
\end{aligned} \tag{14}$$

Stage 6. Solve *simultaneously* the subproblems on the *second-level* subdomains $\xi^{x,y}_{k,l}$, $1 \leqslant k \leqslant K-1$, $1 \leqslant l \leqslant L-1$:

$$\begin{aligned}
&\mathscr{L}\overset{xy}{\psi}{}^{n}_{k,l}(P) - \frac{\overset{xy}{\psi}{}^{n}_{k,l}(P) - W^{n-1}(P)}{\tau} = f(P,t^n), \quad P \in \xi^{xy}_{k,l},\\
&\overset{xy}{\psi}{}^{n}_{k,l}(P) = \overset{x}{\Psi}{}^{n}(P), \quad P \in \partial\xi^{xy}_{k,l} \cap \bar{\Xi}^{x},\\
&\overset{xy}{\psi}{}^{n}_{k,l}(P) = \overset{y}{\Psi}{}^{n}(P), \quad P \in \partial\xi^{xy}_{k,l} \cap \Xi^{y}.
\end{aligned} \tag{15}$$

Stage 7. Stopping criterion: If $t^n = T$, then stop; otherwise go to Stage 1.

Note that here, as in Algorithm A1, the solution at the nth time step, the *continuous* function $W^n(P)$, $P \in \Omega$, (see (10)) is obtained in a single pass of the algorithm.

Convergence and a convergence rate estimate of the three-level Schwarz method was shown in [21]:

Theorem 2. *The three-level Schwarz method (Algorithm* A2*) converges to the solution* $U^n(P)$ *of problem* (2) *with linear rate* $q_{A2} = q_{A1}Q_{A2} \in (0,1)$, *that is*

$$\max_{1 \leqslant n \leqslant N} \left[\max_{P \in \Omega} |U^n(P) - W^n(P)| \right] \leqslant c q_{A2},$$

where q_{A1} *was given in Theorem* 1, $W^n(P)$ *was defined in* (10), *and the constant* c *is independent of* q_{A2}. *For* $Q_{A2} \in (0,1)$ *the following bound holds:*

$$Q_{A2} < 2 \left\{ \exp\left[-\frac{D^x}{\mu \tau^{1/2}} \right] + \exp\left[-\frac{D^y}{\mu \tau^{1/2}} \right] \right\},$$

where D^x *and* D^y *were introduced in Theorem* 1.

Remark 2. Algorithm A2 involves six sequential stages and can be realized on $K \times L$ processors (see also Fig. 2). By its implementation on $(K-1) \times L + K \times (L-1)$ processors (in relation to the total number of the subproblems associated with $\xi^x_{k,l}$, $\xi^y_{k,l}$ or $\theta^x_{k,l}$, $\theta^y_{k,l}$), we can reduce the number of its sequential stages to four. In this case, the subproblems from (9) and (11) (in Stages 1 and 2, respectively) can be solved in parallel. Same holds for the subproblems from (13) and (14) (in Stages 4 and 5, respectively).

2.4. Multi-level Schwarz methods with time extrapolation

In [4], a time extrapolation technique for increasing the convergence rate of the domain decomposition algorithms like A1 and A2 was proposed. The resulting new algorithms differ from the original ones by the determination of the conditions on the inner boundaries for the first-level subproblems, for Algorithm A1, or for the zeroth-level subproblems, for Algorithm A2. The modified boundary conditions are given by the extrapolation in the time variable using the approximate solutions from some previous time steps. Here, we consider the multi-level Schwarz methods with "two-step" extrapolation, and we shall call them Algorithms A1E and A2E. In these algorithms, instead of the solution at previous time step $W^{n-1}(P)$ in (4) and (9), (11), the following expressions are applied:

$$u^0(P) + \tau[\mathscr{L}u^0(P) - f(P,0)], \quad n = 1,$$
$$2W^{n-1}(P) - W^{n-2}(P), \quad n \geqslant 2.$$

Convergence and a convergence rate estimate for the multi-level Schwarz methods with time extrapolation can be found in [21]:

Theorem 3. *If the solution of problem* (1) *is smooth enough, then the multi-level Schwarz methods with time extrapolation (Algorithms* A1E *and* A2E*) converge to the solution* $U^n(P)$ *of problem*

(2) *with linear rate* $q_{AE} \in (0,1)$, *that is*

$$\max_{1 \leqslant n \leqslant N} \left[\max_{P \in \Omega} |U^n(P) - W^n(P)| \right] \leqslant c q_{AE},$$

where $W^n(P)$ *was defined in* (5) (*Algorithm* A1) *or* (10) (*Algorithm* A2), *and the constant* c *is independent of* q_{AE}. *For* q_{AE}, *the following bound holds:*

$$q_{AE} = \frac{1 - (q_A)^N}{(1 - q_A)N} q_A,$$

where by q_A *is meant* q_{A1} *or* q_{A2} *defined in Theorem* 1 *or* 2, *respectively, and* N *is the number of time steps* (*see Section* 2.1).

2.5. Some additional remarks

In this section, we have collected some remarks about the algorithms.

Remark 3. From Lemma 1, Theorems 1, 2 and 3, it follows that Algorithms A1, A1E and A2, A2E, converge to the solution $u(P,t)$ of the continuous problem (1) with the rate $\mathcal{O}(q + \tau)$.

Remark 4. In the description of the algorithms, no attention was given to the choice of a domain partitioning according to the boundary layers. It means that the proposed multi-level Schwarz methods are convergent with any (rectangular) domain decomposition, but in the numerical realization, we can and we should take into account the special properties of the singularly perturbed problems, see Section 3.3.

Remark 5. Here, we shortly discuss the numerical stability of the proposed Schwarz algorithms. Namely, the algorithms were described on differential level, but the numerical realization (implementation) necessarily produces computational errors, since the subproblems of the Schwarz methods cannot be solved exactly. In our case, total computational errors consist of computer-dependent errors (round-off errors) as well as errors determined by a numerical approach to be used.

Let us denote the multi-level Schwarz algorithms (without time extrapolation) perturbed by the computational errors by Algorithm $\widetilde{\text{A1}}$ (two-level Schwarz method) and Algorithm $\widetilde{\text{A2}}$ (three-level Schwarz method). The solution obtained by these "perturbed" algorithms at each time step are also marked with tilde $\tilde{\ }$. Let δ_λ^n denote the maximum absolute value of the computational errors made on the λth level at the nth time step of the "perturbed" multi-level algorithms (either $\widetilde{\text{A1}}$ or $\widetilde{\text{A2}}$), and denote by Δ^n is the maximum absolute value of the differences between the solutions of "perturbed" and "unperturbed" Schwarz algorithms at the nth time step (see (5) or (10)), that is

$$\Delta^n \equiv \max_{P \in \Omega} |\tilde{W}^n(P) - W^n(P)|, \quad 1 \leqslant n \leqslant N.$$

Then, for this value Δ^n, the following estimate was shown in [21]:

$$\max_{1 \leqslant n \leqslant N} \Delta^n \leqslant N \left(\max_{1 \leqslant n \leqslant N} \delta_1^n + \max_{1 \leqslant n \leqslant N} \delta_2^n \right) + N \begin{cases} 0, & \text{for Algorithm } \widetilde{\text{A1}}, \\ q_{A1} \max\limits_{1 \leqslant n \leqslant N} \delta_0^n, & \text{for Algorithm } \widetilde{\text{A2}}, \end{cases}$$

where N is the number of time steps and $q_{A1} \in (0,1)$ was given in Theorem 1.

Note that the similar stability results hold in the case of the multi-level Schwarz algorithms with time extrapolation A1E and A2E.

3. Computer realization

In this section, we consider the computer realization of the Schwarz algorithms on a distributed memory multiprocessor computer Cray T3E (224 Alpha EV5 processors (375 MHz), each having a 128 MB main memory). In our parallel experiments to be reported in Section 4, we shall only use nine processors, and the parallelization is based on the message passing interface (MPI) standard [13].

As the test problem, we consider problem (1) with the following input data:

$$X = 1,\ Y = 1,\ T = 0.1, \quad f(P,t) = 0, \quad u^0(P) = 0, \quad \tilde{u}(P,t) = \sin(\omega t),\ \omega = 100\pi. \tag{16}$$

3.1. Numerical approach

We first describe the "undecomposed" algorithm used for the solution of problem (1). Remind that the time discretization reduces the given parabolic problem (1) to the sequence of elliptic problems (2). In the case of the present test problem, for $\mu \leqslant 1$, these elliptic problems are singularly perturbed and have boundary layers of width $\mathcal{O}(\sigma|\ln\sigma|)$, $\sigma = (2/\omega)^{1/2}\mu$, near $\partial\Omega$, where ω was given in (16) (see, e.g., [5]).

In our experiments, to solve the singularly perturbed elliptic problems from (2) at each time step, we apply the following technique: On the domain Ω, a special nonequidistant mesh Ω_h of Bakhvalov's type [1] is introduced (see Section 3.2). The construction of the mesh rests on the estimates of derivatives of the exact solutions of problems (2) that is, the existence of the boundary layers near $\partial\Omega$. The problems from (2) are discretized by the finite differences: the differential operator $\mathscr{L}$ from (1) is approximated by a standard central scheme on the mesh Ω_h. The resulting difference schemes have the second-order μ-uniform convergence [1]. The produced finite-dimensional problems are solved by the incomplete Choleski conjugate gradient (ICCG) method [12].

The approach described above is also applied to solve the subproblems by the numerical realization of the Schwarz methods. Furthermore, Ω_h is assumed to be compatible with the domain decomposition, that is, the mesh lines are used by domain decomposition as boundaries of the subdomains.

3.2. Computational mesh

Next, we shall take a closer look at the construction of the computational mesh for the model domain $\Omega = (0,1) \times (0,1)$: We introduce the nonequidistant mesh by [1]

$$\Omega_h = \left\{x_i : x_i = \eta\left(\frac{i}{M}\right),\ 0 \leqslant i \leqslant M\right\} \times \left\{y_j : y_j = \eta\left(\frac{j}{M}\right),\ 0 \leqslant j \leqslant M\right\},$$

where an even number M is the number of mesh points in x- and y-directions. The mesh generating function $\eta(\alpha)$ has the following form:

$$\eta(\alpha)=\begin{cases}\chi(\alpha), & \alpha\in[0,a],\\ \chi(a)+(\alpha-a)\left.\dfrac{\mathrm{d}\chi(\alpha)}{\mathrm{d}\alpha}\right|_{\alpha=a}, & \alpha\in\left[a,\frac{1}{2}\right],\\ 1-\eta(1-\alpha), & \alpha\in\left[\frac{1}{2},1\right],\end{cases} \tag{17}$$

where $\chi(\alpha)=-2\sigma\ln(1-4\alpha)$, and $0<a<\frac{1}{4}$ is the root of the equation

$$\left.\frac{\mathrm{d}\chi(\alpha)}{\mathrm{d}\alpha}\right|_{\alpha=a}=\frac{\frac{1}{2}-\chi(a)}{\frac{1}{2}-a},$$

with σ being defined in Section 3.1.

Remark 6. Note that at $\alpha\in[\chi(a),1-\chi(a)]$ the mesh generating function $\eta(\alpha)$ produces a uniform mesh with a step size H. From (17), it follows that H is a decreasing function of μ and $H\to 2/M$ as $\mu\to 0$.

3.3. Domain decomposition

In the numerical experiments, we shall consider the domain decomposition with $K=L=3$, that is, we apply our parallel algorithms on nine processors of the parallel computer. As outlined above, the domain Ω is partitioned into the subdomains compatibly with the lines of the discretization mesh Ω_h. For the multi-level Schwarz methods, we choose the *first-level* subdomains $\Omega_{k,l}$ as follows:

$$x_0^C=0,\quad x_1^C=x_{M/3},\quad x_2^C=x_{2M/3},\quad x_3^C=1,$$
$$y_0^C=0,\quad y_1^C=y_{M/3},\quad y_2^C=y_{2M/3},\quad y_3^C=1.$$

The interface (i.e., *the zeroth-level and second-level*) subdomains are constructed in the following manner:

$$x_1^b=x_{(M/3)-\kappa},\quad x_1^e=x_{(M/3)+\kappa},\quad y_1^b=y_{(M/3)-\kappa},\quad y_1^e=y_{(M/3)+\kappa},$$
$$x_2^b=x_{(2M/3)-\kappa},\quad x_2^e=x_{(2M/3)+\kappa},\quad y_2^b=y_{(2M/3)-\kappa},\quad y_2^e=y_{(2M/3)+\kappa},$$
$$1\leqslant\kappa\leqslant\left[\left(\frac{1}{3}-a\right)M\right],$$

where a was defined in (17) and $[c]$ denotes the truncation of the fractional part of a number c.

As is seen from the above relationships, the presented domain partitioning produce equal numbers of mesh points in the first-level subdomains $\Omega_{k,l}$. Furthermore, the overlapping regions of the first-level subdomains and the interface subdomains are located in the zones of the uniform mesh and can be measured by a number κ of the uniform step size H (see Remark 6), that is, the overlapping interval sizes D^x and D^y (see Section 2) are defined as κH.

Remark 7. We emphasize that the proposed domain partitioning is optimal in order to minimize the common computational cost of the algorithms so far, as it decreases the number of grid points

needed for the zeroth-level subproblems as well as the second-level subproblems, thus, minimizing the computational work for their solution. However, this domain partitioning (the maximum number of the processors) is exclusively determined by the distribution of the mesh points between the boundary layers and other part of the computational domain Ω, in other words, by the structure of the non-equidistant mesh adapted to the behaviour of the exact solution of the original singularly perturbed problem (1). Therefore, the considered numerical realization of our algorithms is suitable for parallel processing with coarse granularity.

3.4. Inexact variants of Algorithms A2 and A2E

Here, we discuss a possibility to reduce the total computational cost of the approximate solution of problem (1) by the above-described numerical realization of the three-level Schwarz methods.

In [21], the so-called inexact variants of Algorithms A2 and A2E were proposed. In this case, the zeroth-level subproblems from (9) and (11) are approximated on coarser mesh than the original mesh Ω_h introduced in Section 3.2. Namely, assuming, for simplicity, that the mesh point number M is divided exactly by a number $m > 1$, for the approximation of the subproblems, we use the following mesh:

$$\Omega_h^{(m)} = \left\{ x_I : x_I = \eta\left(\frac{mI}{M}\right),\ 0 \leqslant I \leqslant \frac{M}{m} \right\} \times \left\{ y_J : y_J = \eta\left(\frac{mJ}{M}\right),\ 0 \leqslant J \leqslant \frac{M}{m} \right\}.$$

As is seen from (17), $\Omega_h^{(m)}$ consists only of a subset of the nodes of the mesh Ω_h, that is

$$x_I = x_i, \quad i = mI, \quad 0 \leqslant I \leqslant \frac{M}{m}, \qquad y_J = y_j, \quad j = mJ, \quad 0 \leqslant J \leqslant \frac{M}{m}.$$

The boundary conditions for the first-level subproblems (approximated on the original mesh Ω_h) are determined from the solutions of the zeroth-level subproblems (given on the coarse mesh $\Omega_h^{(m)}$) by an interpolation formula. For the domain partitioning described in Section 3.3, we may specify the formula as follows:

$$v^n_{h,1,1}(x_{i+v}, y_{M/3}) = \frac{v}{m}\left[\overset{y}{\phi}{}^n_{h,1,1}(x_{I+1}, y_{M/3m})\, \overset{y}{\phi}{}^n_{h,1,1}(x_I, y_{M/3}) \right] + \overset{y}{\phi}{}^n_{h,1,1}(x_I, y_{M/3m}),$$

$$i = mI, \quad 0 \leqslant v \leqslant m, \quad 0 \leqslant I \leqslant \frac{M}{3m} - 1,$$

where $v^n_{h,1,1}(x_i, y_j)$ and $\overset{y}{\phi}{}^n_{h,1,1}(x_I, y_J)$ are the finite-dimensional approximations of the corresponding functions from (12) and (11) given on Ω_h and $\Omega_h^{(m)}$, respectively (see [21,23] for details).

It should be noted that for the inexact variant of Algorithm A2 or A2E, the computational errors made on each time step by solving the zeroth-level subproblems and by the determination of the corresponding boundary conditions for the first-level subproblems are in excess of the errors obtained by the "exact" realization of the algorithms with $m = 1$. Indeed, ignoring round-off errors as well as the residual errors of the ICCG method, we obtain the estimate $\mathcal{O}(m^2/M^2)$ for the above-mentioned computational errors. However, from the numerical stability of the Schwarz methods (see Remark 5) it follows that we can select the values of m under which the computational costs of solving the zeroth-level subproblems are significantly decreased, but the real convergence rate of these algorithms varies only slightly or not at all.

In Section 4, we present experimental results for numerical realizations of Algorithms A2 and A2E at $m = 1, 2, 4$. For these realizations of the algorithms, we introduce the notations: $\mathrm{A2}_{(m)}$ and $\mathrm{A2E}_{(m)}$.

3.5. Parallelization using MPI

Parallelization of the proposed multi-level Schwarz methods on a distributed memory multiprocessor computer is rather straightforward: In all the algorithms, the subproblems related to the first-level subdomains $\Omega_{k,l}$ are mapped onto individual processors. The correcting subproblems (related to $\xi^x_{k,l}$, $\xi^y_{k,l}$ and $\xi^{xy}_{k,l}$), of the all algorithms, as well as the predicting subproblems (related to θ^x_k and θ^y_l), of Algorithms A2 and A2E, are placed into processors containing one of the first-level subproblems (see Figs. 1 and 2).

The parallel implementation of the algorithms is then based on the message passing interface (MPI) library [13]: To distribute the (equal) initial data to all processors, we use the MPI subroutine `MPI_BCAST`. To exchange data between the subproblems located in different processors, we use blocking send and receive MPI subroutines `MPI_BSEND` and `MPI_RECV`. The elapsed wall-clock time was measured by the MPI subroutine `MPI_WTIME`.

4. Numerical results

In this section, we report on the computational experiments made on a distributed memory multiprocessor computer Cray T3E. The multi-level Schwarz methods (Algorithms A1, A1E and A2, A2E) are compared among themselves as well as to the sequential "undecomposed" algorithm, that is, the algorithm, where the original problem (1) is solved approximately by the numerical technique proposed in Section 3.1.

In the numerical experiments, we use the mesh Ω_h with $M = 300$. For the time discretization, we choose the step size $\tau = 10^{-3}$, implying the number $N = T/\tau = 10^2$ of time steps. The ICCG method is finished when accuracy of $\epsilon = 10^{-5}$ is achieved.

First, we present the experimental convergence results for the multi-level Schwarz methods with different values of the perturbation parameter μ and the overlapping interval sizes. Remind that the overlapping interval sizes are measured by the number κ of step size H of the uniform mesh (see Remark 6) and that H is a decreasing function of μ (see Table 1).

Here, as proposed in [20], to establish of experimental convergence properties of the algorithms, we use values of the maximum residual:

$$R_{\max} = \max_{1 \leqslant n \leqslant N} \left[\max_{P \in \Omega} \left| \mathscr{L}_h W_h^n(P) - \frac{W_h^n(P) - W_h^{n-1}(P)}{\tau} - f(P, t^n) \right| \right],$$

instead of those of the maximum error (see Theorems 1,2 and 3):

$$E_{\max} = \max_{1 \leqslant n \leqslant N} \left[\max_{P \in \Omega} |U_h^n(P) - W_h^n(P)| \right],$$

where $\mathscr{L}_h$ is the finite difference discretization of the differential operator $\mathscr{L}$ from (1) (see Section 3.1), $U_h^n(P)$ and $W_h^n(P)$ are the finite-dimensional approximations of the corresponding functions

Table 1
The μ-dependence of κ_ϵ for the Schwarz algorithms (by the overlapping interval sizes $\kappa_\epsilon H$, the accuracy of the corresponding Schwarz algorithm conforms to the accuracy ϵ of the numerical method used for solving the subproblems on each time step). The μ-dependence of the step size H of the uniform mesh is appended for reference

		Algorithm							
μ	$H, \times 10^{-3}$	A1	A1E	$A2_{(1)}$	$A2_{(2)}$	$A2_{(4)}$	$A2E_{(1)}$	$A2E_{(2)}$	$A2E_{(4)}$
10^{0}	3.488	44	37	27	28	28	24	24	24
$10^{-0.5}$	4.688	11	9	7	8	8	6	6	8
10^{-1}	5.752	2	2	1	—	—	1	—	—
$10^{-1.5}$	6.294	1	1	1	—	—	1	—	—
10^{-2}	6.524	1	1	1	—	—	1	—	—

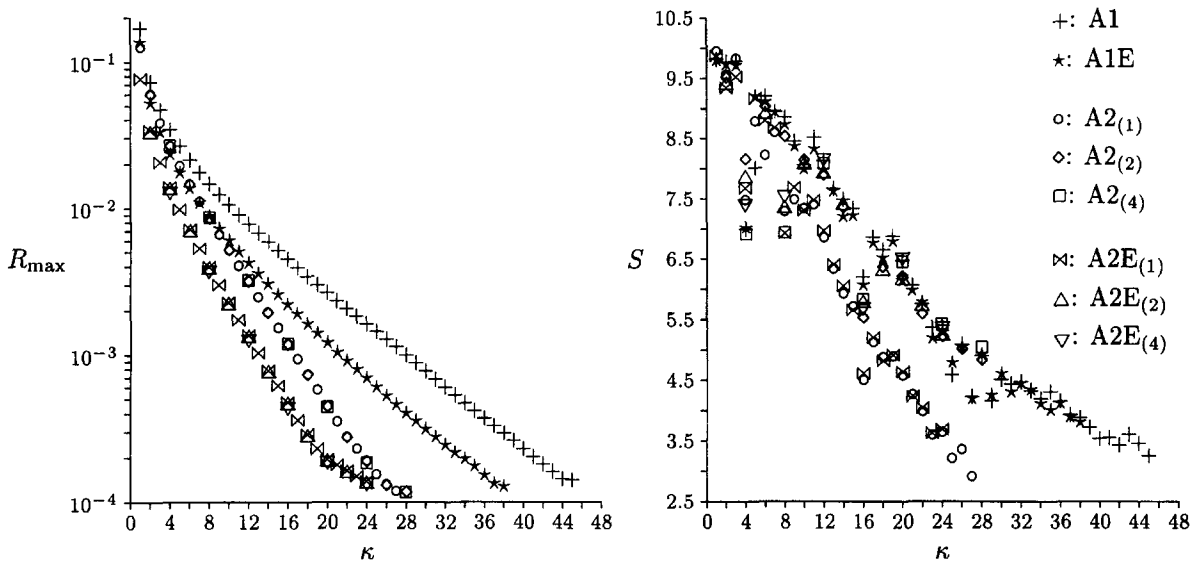

Fig. 3. κ-dependence of the maximum residual R_{max} and the speedups S for the domain decomposition algorithms at $\mu = 1$.

from (2) and(5) or (10), respectively. Note that in this case the comparison with the solution of the "undecomposed" problem is not needed.

In Figs. 3 and 4, we give the κ-dependence of R_{max} at different values of μ. As is seen from the "curves", at fixed μ, the maximum residual R_{max} is a decreasing function of exponential type with respect to the overlapping interval sizes κH. These numerical results for the maximum residual are in good agreement with the theoretical estimates presented in Section 2 for the convergence rates of the maximum error E_{max}.

Furthermore, we can make the following observation from the "curves" $R_{max}(\kappa)$: For each Schwarz algorithm, at given μ, beginning with some number κ_ϵ, the maximum residual varies only slightly, and for it, the bound $R_{max} = \mathcal{O}(r_\epsilon)$ holds, where r_ϵ is the maximum value of the residual errors achieved by the numerical solution of the subproblems (note that r_ϵ is determined by the prescribed accuracy ϵ of the ICCG method). In other words, by $\kappa \geqslant \kappa_\epsilon$ the accuracy of the Schwarz algorithm

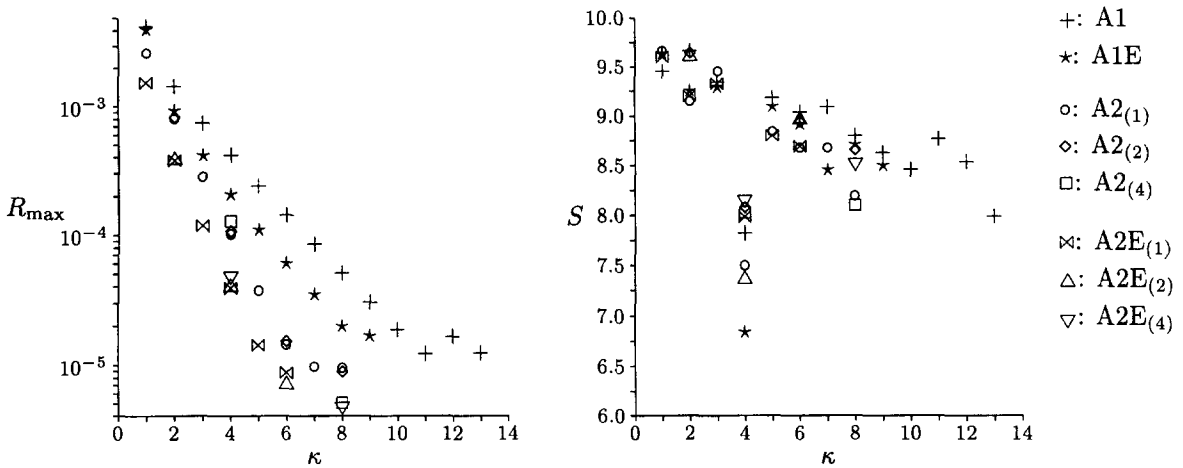

Fig. 4. κ-dependence of the maximum residual R_{max} and the speedups S for the domain decomposition algorithms at $\mu = 10^{-0.5}$.

conforms to that of the numerical method used for solving the subproblems at each time step (see Section 3.1).

Table 1 demonstrates the μ-dependence of κ_ϵ for all examined algorithms. As might be expected from Theorems 1–3, κ_ϵ is an increasing function of the perturbation parameter μ and, moreover, for $\mu \leqslant 10^{-1.5}$, we have $\kappa_\epsilon = 1$ for all algorithms.

Next, we compare the experimental convergence properties of the two-level and three-level Schwarz algorithms. As the data of Figs. 3 and 4 suggests, the experimental convergence rate of Algorithm A2 is about twice as much than that of Algorithm A1. These experimental results are in keeping with the estimates presented in Theorems 1 and 2. We emphasize that the figures demonstrate equal convergence rate of the "exact" realization of Algorithm A2 (at $m=1$) and that of the inexact variant of Algorithm A2 at $m = 2, 4$ (see Section 3.4).

We compare next the convergence rate of the Schwarz algorithms with and without the time extrapolation. Using the results from [21], it can be shown that in the case of the test problem with the input data from (16), by sufficiently small q_A (the convergence rate of Algorithms A1 or A2), the following estimates hold: $q_A/q_{AE} = \mathcal{O}(\tau^{-1}\omega^{-1})$, where q_{AE} denotes the convergence rate of the Schwarz algorithms with the "two-step" time extrapolation A1E or A2E (see Theorems 1, 2 and 3), and ω was defined in (16). These theoretical expressions are supported by the experimental results presented in Figs. 3 and 4 (note that in the experiments, we have $\tau\omega = 0.1\pi$). Finally, we remark that the distinctions between the convergence rates of the examined Schwarz algorithms (A1, A1E, A2 and A2E) entail the differences in the μ-dependence of κ_ϵ for these algorithms (see Table 1).

Next, we evaluate computational effectiveness of the Schwarz algorithms. In Figs. 3 and 4 as well as in Table 2, we give the speedups for Algorithms A1, A1E and A2, A2E with respect to the sequential ("undecomposed") algorithm. We here define the speedup (of the parallel algorithm compared to the corresponding sequential one) by the notation $S = T_s/T_p$, where T_s is the execution time for the sequential algorithm and T_p for the domain decomposition algorithms via parallel processing.

Table 2
The μ-dependence of the speedups S for the parallel implementation of the multi-level Schwarz methods at $\kappa = \kappa_\epsilon$ (see also Table 1)

	Algorithm							
μ	A1	A1E	$A2_{(1)}$	$A2_{(2)}$	$A2_{(4)}$	$A2E_{(1)}$	$A2E_{(2)}$	$A2E_{(4)}$
10^{0}	3.47	3.90	2.90	4.83	5.01	3.66	5.23	5.39
$10^{-0.5}$	8.77	8.50	8.67	8.65	8.08	8.67	8.96	8.53
10^{-1}	9.91	9.62	10.01	—	—	9.91	—	—
$10^{-1.5}$	9.88	10.08	10.06	—	—	10.16	—	—
10^{-2}	9.98	10.17	10.17	—	—	10.25	—	—

The Figs. 3 and 4 demonstrate the κ-dependence of the speedups S for the algorithms with different μ. The figures indicate that there is a significant difference in the behaviour of the "curves" $S(\kappa)$ for Algorithms A1 and $A2_{(1)}$ (the "exact" realization of Algorithm A2 at $m = 1$). This is because the realization of Algorithm A2 on each time step consists two additional sequential stages for solving the zeroth-level subproblems. The implementation of the inexact variant of Algorithm A2 at $m > 1$ decreases the computational cost of these stages. One can see from Fig. 3 that, even at $m = 2$, the κ-dependence of the speedups for the inexact variant of Algorithm A2 differs little from that for Algorithm A1. Note that the κ-dependence of the speedups for Algorithms $A2_{(m)}$ is moderately affected as m increases from 2 to 4. From the figures, we can conclude that the "curves" $S(\kappa)$ for the Schwarz algorithms with and without the time extrapolation are close to each other.

We emphasize that the κ-dependence of the speedups are distinctly perturbed by a nonlinear dependence of the execution time on the dimensions of the discrete zeroth-level and second-level subproblems. Probably, there are sets of "optimal" as well as "nonoptimal" dimensions of processed vectors determined by the size of the cache-memory of the processors. This is supported by the existence of "peaks" and "pits" with μ-independent positions in the "curves" $S(\kappa)$ (compare Figs. 3 and 4).

In Table 2, the values of the speedups $S(\kappa_\epsilon)$ for the algorithms at different μ are listed. First of all, we emphasize that $S(\kappa_\epsilon)$ is a decreasing function of the perturbation parameter μ, because κ_ϵ increases with respect to μ. Furthermore, the results confirm that, basically, the parallel implementation of Algorithm $A2_{(m)}$ by a reasonable choice of the value m is faster than that of Algorithm A1. From the table, it also follows that the computational effectiveness of the Schwarz algorithms can be appreciably affected by the time extrapolation. Note that perturbation of these relations (e.g., in the case of Algorithms A1 and $A2_{(m)}$ at $\mu = 10^{-0.5}$ as well as for the Schwarz algorithms with and without the time extrapolation at $\mu = 10^{-1}$) are exclusively related to the above-mentioned "cash-memory" effect.

Finally, we discuss the efficiency of our algorithms which is defined by $S(\kappa_\epsilon)/p$, where $p=9$ is the number of the processors used in parallel. From Table 2, it follows that at $\mu \leqslant 0.1$, the efficiency of the algorithms exceeds one. Most likely, this is explained by two reasons: First of all, the computational costs of the Schwarz algorithms (calculated from the total number of the ICCG iterations required for solving the subproblems and the number of mesh points in the corresponding subdomains) prove to be less than that of the "undecomposed" algorithm in the case of the original problem (2). The second reason is the effect of the cache-memory of the processors used in computations (as in

the case of the "peaks" in the "curves" $S(\kappa)$): frequently accessed vectors in the subproblems in each processor fit better to the cache-memory than those of the "undecomposed" problem in one processor and, hence, the total amount of cache misses is reduced in the parallel case.

In closing, we point out that in our experiments typical times used for communication between processors were under 1% of the total (parallel) execution times. It means that the examined algorithms are well suited for distributed memory computing from the point of view of the communication costs.

5. Conclusions

We have compared the parallel implementation of the two-level Schwarz method with the correcting subproblems [3,20] as well as the three-level Schwarz method with the predicting and correcting subproblems [21]. Moreover, we have examined the modification of these methods using the "two-step" time extrapolation on subdomain interfaces [4,20,21].

The experimental results substantiate a priori estimates of the convergence rate for the multi-level Schwarz methods with respect to the perturbation parameter and the overlapping interval sizes. It has been demonstrated that the computational effectiveness of the "inexact" variant of the three-level Schwarz method with time extrapolation is superior to that of all other examined methods.

Acknowledgements

The research was done during the first author's visit at the Laboratory of Scientific Computing of University of Jyväskylä (September–October, 1997); the visit was supported by the Academy of Finland (Grant No. 34063). The numerical experiments were carried out on a Cray T3E computer, located in Center for Scientific Computing, Espoo, Finland.

References

[1] N. Bakhvalov, On optimization of methods for solving boundary value problems in the presence of boundary layers, Zh. Vychisl. Mat. Mat. Fiz. 9 (1969) 841–859 (in Russian).

[2] I. Boglaev, V. Sirotkin, Domain decomposition technique for singularly perturbed problems and its parallel implementation, in: J. Miller, R. Vichenevetsky (Eds.), Proceedings of 13th IMACS World Congress on Computation and Applied Mathematics, IMACS, Dublin, 1991, pp. 522–523.

[3] I. Boglaev, V. Sirotkin, Iterative domain decomposition algorithms for the solution of singularly perturbed parabolic problems, Comput. Math. Appl. 31 (1996) 83–100.

[4] I. Boglaev, V. Sirotkin, Solution of a singularly perturbed parabolic problem by domain decomposition methods with time extrapolation, Comput. Math. Appl. 32 (1996) 75–90.

[5] I. Boglaev, V. Sirotkin, J. Lavers, The computation of transient 2-D eddy current problem by domain decomposition algorithms, Math. Comput. Modelling 21 (1995) 39–51.

[6] X.-C. Cai, Multiplicative Schwarz algorithms for parabolic problems, SIAM J. Sci. Comput. 15 (1994) 587–603.

[7] M. Garbey, A Schwarz alternating procedure for singular perturbation problems, SIAM J. Sci. Comput. 17 (1996) 1175–1201.

[8] Yu. Kuznetsov, Overlapping domain decomposition methods for FE-problems with elliptic singular perturbed operators, in: R. Glowinski, Yu. Kuznetsov, G. Meurant, J. Periaux, O. Widlund (Eds.), Fourth International

Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1991, pp. 223–241.

[9] Yu. Kuznetsov, Overlapping domain decomposition methods for parabolic problems, in: Yu. Kuznetsov, J. Periaux, A. Quarteroni, O. Widlund (Eds.), Domain Decomposition Methods in Science and Engineering, AMS, Providence, RI, 1994, pp. 63–70.

[10] O. Ladyženskaja, V. Solonnikov, N. Ural'ceva, Linear and Quasi-linear Equations of Parabolic Type, AMS, Providence, RI 1968.

[11] P. Lions, On the Schwarz alternating method I, in: R. Glowinski, G. Golub, G. Meurant, J. Periaux (Eds.), First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1988, pp. 2–42.

[12] J. Meijerink, H. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comput. 31 (1977) 148–162.

[13] Message Passing Interface Forum, MPI: a message-passing interface standard, Tech. Rep. CS-94-230, Computer Science Dept., University of Tennessee, Knoxville, TN, 1994.

[14] G. Rodrigue, E. Reiter, An asymptotically induced domain decomposition method for parabolic boundary layer problems, in: H. Kaper, M. Garbey (Eds.), Asymptotic Analysis and the Numerical Solution of Partial Differential Equations, Lecture Notes in Pure and Appl. Math., Dekker, New York, 1991, pp. 55–73.

[15] A. Samarskii, Theory of Finite Difference Schemes, Nauka, Moscow, 1989 (in Russian).

[16] H. Schwarz, Über einige Abbildungsaufgaben, Gesammetter. Math. Abh. 11 (1869) 65–83.

[17] J. Scroggs, Physically motivated domain decomposition method for singularly perturbed equations, SIAM J. Numer. Anal. 28 (1991) 168–178.

[18] G. Shishkin, Grid approximation of singularly perturbed boundary value problem for quasi-linear parabolic equations in the case of complete degeneracy in spatial variables, Soviet J. Numer. Anal. Math. Modelling 6 (1991) 243–261.

[19] V. Sirotkin, The solution of singularly perturbed semi-linear elliptic problem by iterative domain decomposition algorithms, Comput. Math. Appl. 33 (1997) 99–116.

[20] V. Sirotkin, The solution of singularly perturbed parabolic problems by parallel algorithms combining Crank-Nicholson scheme and overlapping Schwarz methods, Comput. Math. Appl., to appear.

[21] V. Sirotkin, Multi-level Schwarz methods for parallel solving singularly perturbed parabolic problems, in preparation

[22] V. Sirotkin, P. Tarvainen, Parallel numerical algorithms based on overlapping Schwarz methods for singularly perturbed semi-linear elliptic problems, Tech. Rep. 24, University of Jyväskylä, Department of Mathematics, Laboratory of Scientific Computing, 1996.

[23] V. Sirotkin, P. Tarvainen, Parallel multi-level Schwarz methods for singularly perturbed parabolic problems, Tech. Rep. 19, University of Jyväskylä, Department of Mathematics, Laboratory of Scientific Computing, 1997.

[24] V. Sirotkin, P. Tarvainen, Overlapping Schwarz methods for singularly perturbed semi-linear elliptic problems and their parallel implementation, SIAM J. Sci. Comp., to appear.